

# Инструкции по доработке конфигурации

## Отладка через конфигуратор 1С

Для запуска отладки конфигурации из конфигуратора 1С нужно указать параметры сеанса 1С (указать используемый номер ПОС):

```
STARTER+POSNUMBER="1"+ROCKEY
```

Если используется ключ катран, тогда вместо ROCKEY нужно написать KATRAN.

## Возможные способы доработки конфигурации:

Рекомендуем ознакомиться с вебинаром [Вебинар - доработки конфигурации Кассир 5](#)

1. пользовательские раскладки клавиатуры
2. [пользовательские кнопки в форме супервизор](#)
3. [расширенные отчеты](#)
4. модуль логики
5. изменение конфигурации и типовых внешних обработок

Способ №1 (использование собственных раскладок клавиатуры) рекомендуется использовать когда клиенту нужна функция (кнопка или горячая клавиша), которая выполнит типовые действия, которые отсутствуют в предопределенных раскладках клавиатуры. Например, применить определенную (или из списка) дисконтную карту, товар, продавца и т.п. Т.е. использовать набор функций формы регистрации.

Способ №2 используется когда нужно добавить дополнительную кнопку в супервизор.

Способ №3: Расширенные отчеты используются для вывода какой-то информации на экран или принтер, могут вызываться пользователем в произвольный момент времени и, опционально, автоматически при снятии отчета с гашением

[Способ №4 решает большинство задач без необходимости доработок конфигурации и типовых обработок с максимальной совместимостью при обновлении конфигурации.](#)

[API логических модулей](#)

[Список обработчиков в защищенной компоненте](#)

Ниже можно скачать архивы с заготовкой модуля логики. В конце модуля основной формы каждой заготовки есть краткая пошаговая инструкция.

ПустаяЗаготовка

- пустой модуль логики, обработчики для добавления закладки подключатся автоматически при заполнении переменных `ИмяМодуля` и `ЗаголовокЗакладкиНастроек`.

ЗаготовкаМакетЧека

- заготовка модуля логики, который позволяет добавить дополнительные поля в макет чека.

ЗаготовкаОсновныеОбработчики

- заготовка модуля с наиболее часто используемыми обработчиками. Каждый обработчик

[прокомментирован.](#)

## Переопределение активности автоскидок

- модуль логики позволяет программно переопределять активность автоматических скидок.

Вариант №5 крайне не рекомендуется использовать по следующим причинам:

- Изменение базовой версии конфигурации невозможно.
- Обновление через дистрибутив невозможно, т.к. дистрибутив выполняет загрузку оригинальной конфигурации.
- При каждом обновлении конфигурации потребуются переносить изменения вручную.
- В новой версии конфигурации возможно изменение кода, которое потребует дополнительное время на актуализацию доработок.

## Основные реквизиты, функции, процедуры,

При доработке конфигурации рекомендуется использовать типовые методы, которые можно найти в коде конфигурации.

Список методов в защищенной компоненте описан в статье [API компоненты AddIn.CashContext](#)

Наиболее часто используемые методы описаны ниже.

Формат описания методов:

Имя метода - описание метода.

Параметры:

Номер параметра: Тип параметра (значение по-умолчанию) - описание параметра.

Пример кода.

### Диалоги

**ИнтерфОшибка** - процедура выводит сообщение пользователю.

Параметры:

1: Строка - строка с текстом сообщения

2: Строка («Ошибка») - заголовок формы

```
ИнтерфОшибка("Текст сообщения", "Внимание!");
```

**ИнтерфВопросДаНет** - глобальная функция открывает окно с переданным текстом и двумя кнопками «Да» и «Нет», возвращает нажатую кнопку («Да» или «Нет»).

Параметры:

1: Строка - строка с текстом вопроса

2: Строка («Нет») - кнопка по-умолчанию («Да» или «Нет»)

```
Если ИнтерфВопросДаНет("Текст вопроса") = "Да" Тогда  
    // обработка ответа на вопрос  
КонецЕсли;
```

**ИнтерфВопросДаНетОтмена** - глобальная функция открывает окно с переданным текстом и кнопками «Да», «Нет», «Отмена», при нажатии кнопки возвращает строку («Да», «Нет»,

«Отмена»).

Параметры:

1: Строка - строка с текстом вопроса

```
Если ИнтерфВопросДаНетОтмена("Текст вопроса") = "Да" Тогда
    // обработка ответа на вопрос
КонецЕсли;
```

**ПолучитьКоличество, ПолучитьСекцию, ПолучитьСумму и др.** - функции формы Регистрация.ФормаЗапросКоличества открывают окно для ввода числового значения (все возможные функции и их параметры см. в модуле Регистрация.ФормаЗапросКоличества).

```
ФормаЗапросКоличества =
МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.ФормаЗапросКоличества");
Если ФормаЗапросКоличества = Неопределено Тогда
    ФормаЗапросКоличества =
МенеджерОбъектов.СоздатьОбщуюФорму("Регистрация.ФормаЗапросКоличества",
"MAIN" );
КонецЕсли;
НеДробноеЧисло = ФормаЗапросКоличества.ПолучитьЧисло(10, 0, 2); // ввод недробного
двузначного числа, значение по-умолчанию 10
```

**ВыбратьДату, ВыбратьВремя, ВыбратьДатуИВремя** - функции общей формы ВыборДаты открывает форму для ввода даты/времени/даты и времени, тип возвращаемого значения Булево.

1: Произвольный - в эту переменную запишется дата, которую указал пользователь.

2: Дата (Неопределено) - значение по-умолчанию.

```
УИД_Формы = Новый УникальныйИдентификатор();
ФормаВыбораДаты = МенеджерОбъектов.СоздатьОбщуюФорму("ОбщаяФорма.ВыборДаты",
УИД_Формы);
Результат = Неопределено;
ДатаРождения = Дата(2000, 1, 1);

Если ФормаВыбораДаты.ВыбратьДату(Результат, ДатаРождения) Тогда
    ДатаРождения = Результат;
КонецЕсли;

МенеджерОбъектов.ВыгрузитьОбщийОбъект("ОбщаяФорма.ВыборДаты", УИД_Формы);
```

## Работа с составом чека

Как получить форму регистрации в коде:

```
ФормаРегистрации = МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.Форма");
```

Как получить состав чека формы регистрации:

```

ФормаРегистрации = МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.Форма");
ТекущаяСтрокаСоставаЧека = ФормаРегистрации.элтСоставЧека.ТекущаяСтрока;

```

```

ФормаРегистрации = МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.Форма");
ЗК = МенеджерОбъектов.ПолучитьОбщийОбъект("AddIn.CashContext");
ЗК.ДобавитьПозициюПоКодуАртикулу(100, "Код", "", -1, ФормаРегистрации);// добавить товар
с кодом 100, цена цена по-умолчанию (из БД)
ЗК.ДобавитьПозициюПоКодуАртикулу(200, "Код", "", 500, ФормаРегистрации);// добавить
товар с кодом 200 с ценой 500
ЗК.ДобавитьПозициюПоКодуАртикулу(300, "Код", "12345", -1, ФормаРегистрации);// добавить
единицу с кодом "12345" товара с кодом 300, цена по-умолчанию

```

```

ФормаРегистрации = МенеджерОбъектов.ПолучитьОбщийОбъект("Регистрация.Форма");
ЗК = МенеджерОбъектов.ПолучитьОбщийОбъект( "AddIn.CashContext" ) ;
ЗК.ДобавитьПозициюПоШтрихкоду("12345", -1, ФормаРегистрации, Ложь, Ложь);
ЗК.ДобавитьПозициюПоШтрихкоду("12345",// штрихкод
                                -1, // цена (-1 = цена из БД)
                                ФормаРегистрации,
                                Ложь,// Истина - штрихкод считан сканером, Ложь - вручную
                                Ложь);// Истина - проверять право для регистрации по штрихкоду, Ложь -
не проверять

```

```
УстановитьГлавноеПолеВвода(""); // сбросить ГПВ
УстановитьГлавноеПолеВвода("Art1"); // установить строку Art1
УстановитьГлавноеПолеВвода(1000.50); // установить строку 1000.5
```

Printed on 2026/02/11 18:31

- 0 - Строка - Получаем строковое выражение Главного поля ввода (при вводе числа - точка не будет присутствовать в строке)
- 1 - Сумма - Получить значение главного поля ввода, преобразованного к формату суммы
- 2 - Количество - Получить значение главного поля ввода, преобразованного к формату количества

Пример:

```
УстановитьГлавноеПолеВвода(1.456);// установить значение ГПВ "1.456"  
Значение = ПолучитьГлавноеПолеВвода(0);// вернёт строку "1456"  
Значение = ПолучитьГлавноеПолеВвода(1);// вернёт число 1.46  
Значение = ПолучитьГлавноеПолеВвода(2);// вернёт число 1.456
```

## Работа с ККМ

Пример печати произвольного текста на ККМ (из формы регистрации, считаем что ККМ подключен):

```
//ф-я КонвертироватьВМассивДляПечати принимает на вход макет чека, можно использовать  
тэги форматирования  
//<CENTER> - выравнивание по центру  
//<LEFT> - выравнивание по левому краю (по-умолчанию)  
//<RIGHT> - выравнивание по правому краю  
//Слева<HR> </HR>Справа - разделить слова пробелами на всю ширину  
//<HR>.</HR> - повторить на всю ширину  
//<s> - штрихкод ean13  
//<s2d> - qr-код  
//<CUT> - отрезка  
ПС = Символ(182);  
Текст = "<CENTER>Строка по центру" + ПС +  
        "Строка слева" + ПС +  
        "<RIGHT>Строка справа" + ПС +  
        "<HR>*</HR>" + ПС +  
        "<s>1234567890123</s>" + ПС +  
        "<s2d>http://host</s2d>" + ПС +  
        " " + ПС +  
        " " + ПС +  
        "<CUT>";  
ЗК = МенеджерОбъектов.ПолучитьОбщийОбъект ("AddIn.CashContext");  
МассивДляПечати = ЗК.КонвертироватьВМассивДляПечати(Текст);  
СтрокаККМ = Кассир5_DataAccess.ПолучитьККМ();  
РезультатПечати_бул = ЗК.FnKKM_PrintStringBlock(СтрокаККМ, МассивДляПечати);
```

## Подключение к ККМ

Если неизвестно подключен ККМ в данный момент или нет, то нужно выполнить следующий код. Если ККМ не подключен, то выполнится подключение, а после окончания работы с ККМ - отключение, если он не был подключен.



1: Число - код временной транзакции для поиска.

```
Транз = Кассир5_DataAccess.ПолучитьВремТранзакцию(100); // получить врем. транзакцию  
с кодом 100
```

**ПолучитьВремТранзакцииПоТипу** - ф-я выполняет отбор временных транзакций по типу (поле ТипТранзакции), тип возвращаемого значения Таблица значений.

```
Транзакции11 = Кассир5_DataAccess.ПолучитьВремТранзакцииПоТипу(11); // в переменную  
Транзакции11 установится таблица со всеми транзакциями с типом 11
```

**ПисатьВремТранзакцию** - ф-я создаёт новый элемент справочника Временные транзакции. Параметры:

1: Число - код транзакции (используется для уникальности врем. транзакций и определяет порядок записи в справочник Транзакции). Если передано значение «0», тогда установится текущий максимальный код + 1.

2: Число - поле Тип транзакции

3: Дата (Неопределено) - поле Дата транзакции (если не указана - подставляется текущая дата)

4: Число - поле Секция

5: Строка (до 150 символов) - поле Код товара (поле используется для записи строковых данных)

6: Число - поле Цена

7: Число - поле Количество

8: Число - поле Сумма

9: Число - номер ККМ (определяет к какому ККМ относится транзакция, если задать несуществующий номер ККМ - транзакция не будет перенесена в справочник Транзакции)

10: Строка («»)- поле Скидка, поле используется для оперативной работы с длинными строками, в справочник Транзакции не переносится

Пример записи врем. транзакции с типом 777 на чек, с явным указанием кода:

```
КодВремТранзакции = 900000;  
// запишем транзакцию на чек, если транзакция с переданным кодом уже существует, тогда  
будет выполнено изменение транзакции  
// дата транзакции = ТекущаяДата(), секция = 1, код товара = "2", цена = 3, количество = 4,  
сумма = 5, номер ккм = по-умолчанию  
Кассир5_DataAccess.ПисатьВремТранзакцию(КодВремТранзакции, 777, ТекущаяДата(), 1,  
"2", 3, 4, 5, Кассир5_DataAccess.ИспользоватьККМ, "");
```

Пример записи врем. транзакции на чек, без явного указания кода:

```
ТипТранзакцииДопИнфо = 777;  
// удаляем транзакцию с типом 777, чтобы не было дублей  
Для Каждого Транз Из  
Кассир5_DataAccess.ПолучитьВремТранзакцииПоТипу(ТипТранзакцииДопИнфо) Цикл  
Кассир5_DataAccess.УдалитьВремТранзакцию(Транз.Код);  
КонецЦикла;  
// запишем транзакцию на чек, код установится автоматически, дата транзакции =  
ТекущаяДата(), секция = 1, код товара = "2", цена = 3, количество = 4, сумма = 5, номер ккм  
= по-умолчанию
```

```
Кассир5_DataAccess.ПисатьВремТранзакцию(0, ТипТранзакцииДопИнфо, ТекущаяДата(), 1,
"2", 3, 4, 5, Кассир5_DataAccess.ИспользоватьККМ, "");
```

См. также в модуле обработки Кассир5 DataAccess:

## ИзменитьВремТранзакциюПоСтруктуре

## ИзменитьВремТранзакцию

### УдалитьВремТранзакцию

## ПолучитьВыборкуСкидок

## ПолучитьВремТранзакцииЧека

## Работа со справочником Транзакции

Для записи транзакции напрямую в справочник Транзакции (транзакции, не относящиеся к конкретному чеку) используется процедура модуля обработки Кассир5 DataAccess

## Записать Транзакции.

### Параметры:

1: Таблица значений (неопределено) - таблица значений, получаемая функцией модуля формы регистрации или модуля обработки Кассир5 **ДобавитьТранзакциюВТаблицу**.

Пример записи одной транзакции с типом 777:

```
Кассир5_DataAccess.ЗаписатьТранзакции(Кассир5_DataAccess.ДобавитьТранзакциюВТаблицу(Неопределено,
Неопределено,// код кассира
777,// тип транзакции
Неопределено,// дата транзакции
0,//секция
"",// код товара
0,//цена
0,//количество
0));//сумма
```

Пример записи нескольких транзакций:

```

ТаблицаТранзакций = Неопределено;
Кассир5_DataAccess.ДобавитьТранзакциюВТаблицу(ТаблицаТранзакций, Неопределено,
777, Неопределено, 0, "", 0, 0, 0);
Кассир5_DataAccess.ДобавитьТранзакциюВТаблицу(ТаблицаТранзакций, Неопределено,
778, Неопределено, 0, "", 0, 0, 0);
Кассир5_DataAccess.ЗаписатьТранзакции(ТаблицаТранзакций);

```

## Работа с правами пользователя

**ПравоДоступно** - функции модуля обработки Кассир5\_DataAccess, выполняет проверку наличия права у пользователя, тип возвращаемого значения булево.

1: Строка - имя права.

2: Булево (Истина) - если у текущего пользователя право отсутствует, но существует пользователь с набором прав, в котором это право присутствует - тогда откроется форма для считывания карты пользователя с этим правом.



- 3: Булево (Истина) - выводить сообщение об отсутствии права.
- 4: Число (Неопределено) - если параметр указан, тогда будет проверяться пользователь с кодом, указанным в этом параметре, иначе текущий пользователь.
- 5: Число (Неопределено) - в этот параметр запишется код пользователя, который подтвердил право.

```
Если НЕ Кассир5_DataAccess.ПравоДоступно("Право", Ложь, Ложь) Тогда  
// действие при отсутствии права без запроса карты и без вывода сообщения  
КонецЕсли;
```

**ПравоОтсутствует** - функции модуля обработки Кассир5\_DataAccess, выполняет проверку отсутствия права у пользователя, тип возвращаемого значения булево.

- 1: Строка - имя права.
- 2: Булево (Истина) - если у текущего пользователя право присутствует, но существует пользователь с набором прав, в котором это право отсутствует - тогда откроется форма для считывания карты пользователя без этого права.
- 3: Булево (Истина) - выводить сообщение при наличии права.
- 4: Число (Неопределено) - если параметр указан, тогда будет проверяться пользователь с кодом, указанным в этом параметре, иначе текущий пользователь.

```
Если НЕ Кассир5_DataAccess.ПравоДоступно("Право", Ложь, Ложь) Тогда  
// действие при отсутствии права без запроса карты и без вывода сообщения  
КонецЕсли;
```

## Отладка

Большинство действий пользователя (нажатие клавиш, события сканера и ридера магнитных карт) обрабатывается в конфигурации как внешнее событие.

Иногда нужно проверить - правильно ли отработает программа при определенных входящих данных от оборудования, а возможности использовать реальное оборудования нет.

Для этого можно (в модуле логики или назначить на кнопку) эмулировать внешнее событие методом GenerateEvent компоненты AddIn.ShtrihContext:

Параметры:

- 1: название источника события (для большинства событий не анализируется)
- 2: название события
- 3: данные события

Пример:

эмуляция работы со сканером:

```
МенеджерОбъектов.ПолучитьОбщийОбъект("AddIn.ShtrihContext").GenerateEvent("Scanner", "ScannerEvent", "123");
```

эмуляция работы с ридером магнитных карт:

```
МенеджерОбъектов.ПолучитьОбщийОбъект("AddIn.ShtrihContext").GenerateEvent("MSCReaderEvent", "MSCReaderEvent", "123");
```

Также в настройках сканера можно добавить эмулятор сканера.

